William Stallings
Data and Computer Communications
7th Edition

Chapter 21
Network Security

Security Requirements

- Confidentiality
- Integrity
- Availability

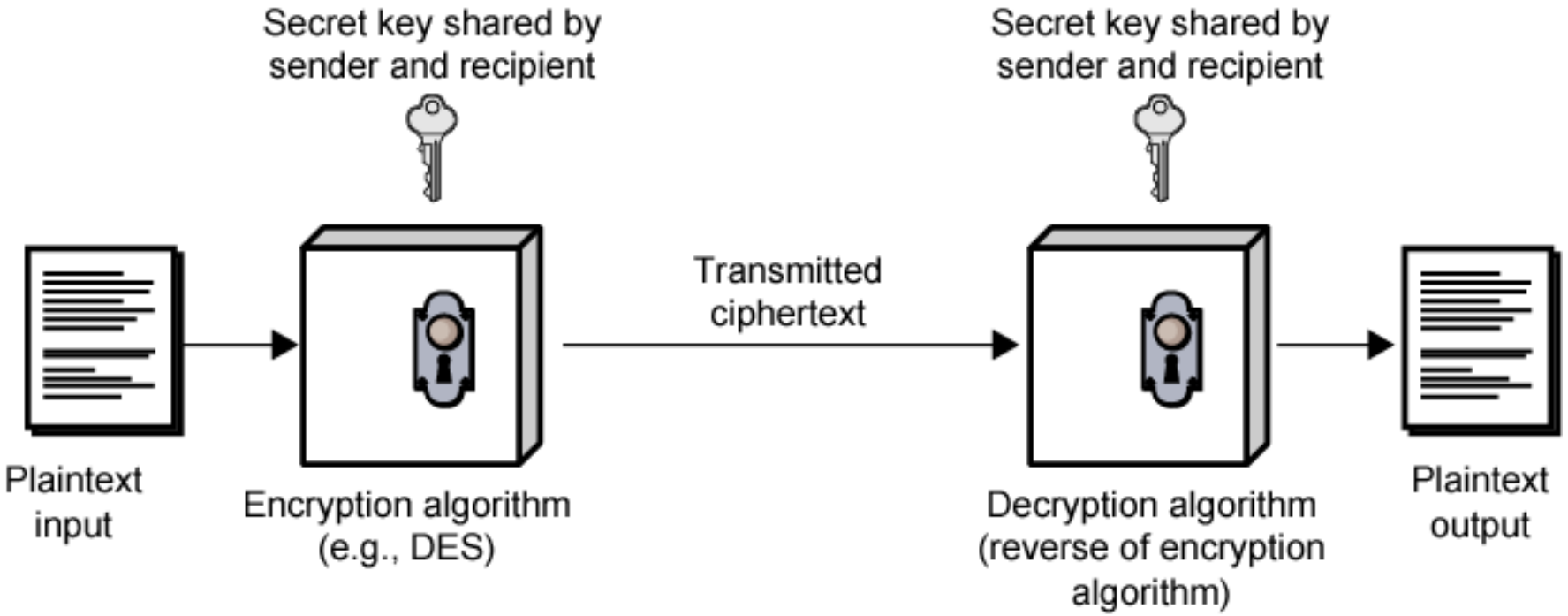
Passive Attacks

- Eavesdropping on transmissions
- To obtain information
- Release of message contents
  - —Outsider learns content of transmission
- Traffic analysis
  - —By monitoring frequency and length of messages, even encrypted, nature of communication may be guessed
- Difficult to detect
- Can be prevented

Active Attacks

- Masquerade
  - —Pretending to be a different entity
- Replay
- Modification of messages
- Denial of service
- Easy to detect
  - —Detection may lead to deterrent
- Hard to prevent

Symmetric Encryption (Simplified)



Ingredients

- Plain text
- Encryption algorithm
- Secret key
- Cipher text
- Decryption algorithm

Requirements for Security

- Strong encryption algorithm
  - —Even if known, should not be able to decrypt or work out key
  - —Even if a number of cipher texts are available together with plain texts of them
- Sender and receiver must obtain secret key securely
- Once key is known, all communication using this key is readable

Attacking Encryption

- Crypt analysis
  - —Relay on nature of algorithm plus some knowledge of general characteristics of plain text
  - —Attempt to deduce plain text or key
- Brute force
  - —Try every possible key until plain text is achieved

Algorithms

- Block cipher
  - —Process plain text in fixed block sizes producing block of cipher text of equal size
  - —Data encryption standard (DES)
  - —Triple DES (TDES)
  - —Advanced Encryption Standard

Data Encryption Standard

- US standard
- 64 bit plain text blocks
- 56 bit key
- Broken in 1998 by Electronic Frontier Foundation
  - —Special purpose machine
  - —Less than three days
  - —DES now worthless

Triple DEA

- ANSI X9.17 (1985)
- Incorporated in DEA standard 1999
- Uses 3 keys and 3 executions of DEA algorithm
- Effective key length 112 or 168 bit
- Slow
- Block size (64 bit) too small

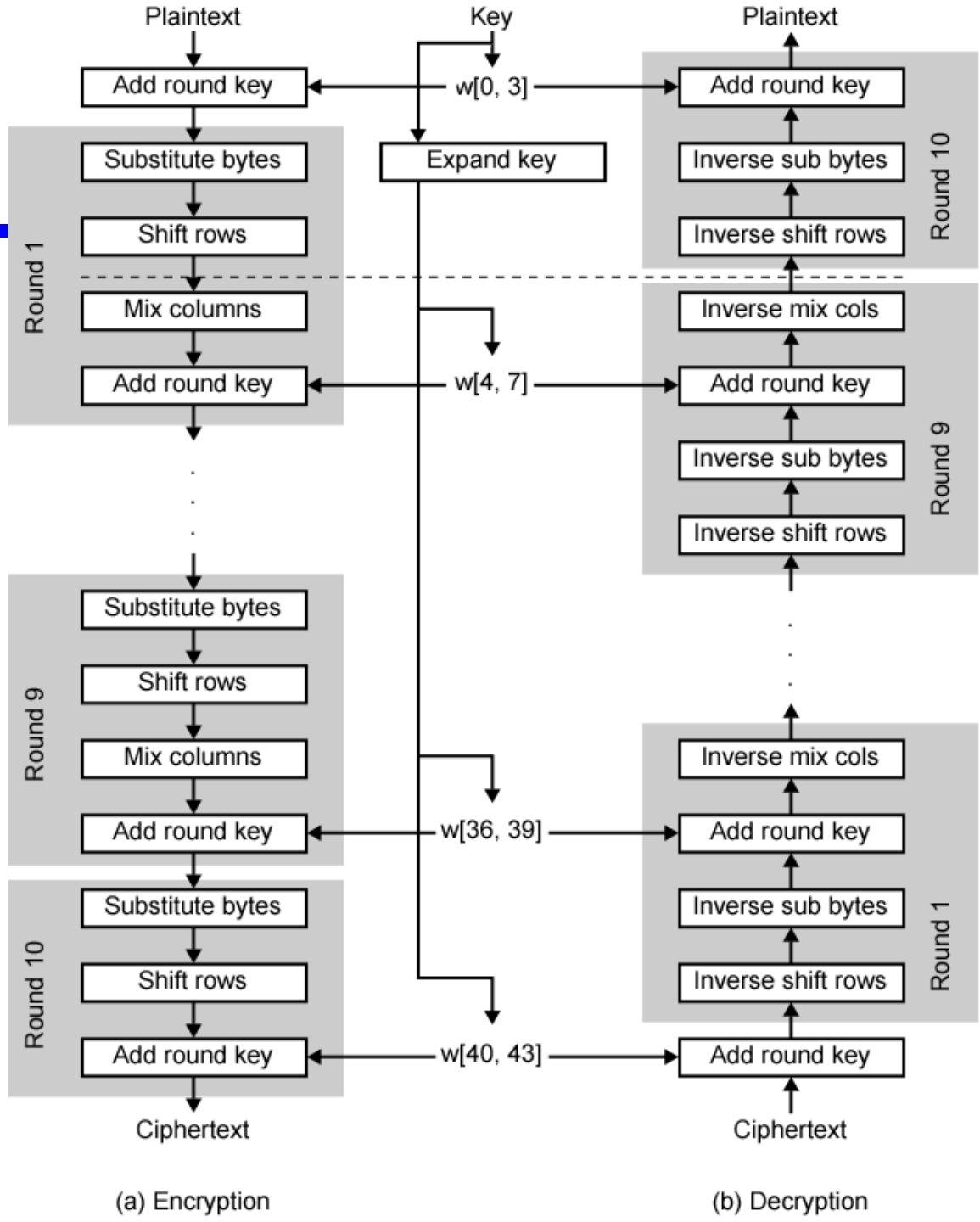
Advanced Encryption Standard

- National Institute of Standards and Technology (NIST) in 1997 issued call for Advanced Encryption Standard (AES)
  - Security strength equal to or better than 3DES
  - Improved efficiency
  - Symmetric block cipher
  - Block length 128 bits
  - Key lengths 128, 192, and 256 bits
  - Evaluation include security, computational efficiency, memory requirements, hardware and software suitability, and flexibility
  - 2001, AES issued as federal information processing standard (FIPS 197)

AES Description

- Assume key length 128 bits
- Input is single 128-bit block
  - Depicted as square matrix of bytes
  - Block copied into State array
    - Modified at each stage
  - After final stage, State copied to output matrix
- 128-bit key depicted as square matrix of bytes
  - Expanded into array of key schedule words
  - Each four bytes
  - Total key schedule 44 words for 128-bit key
- Byte ordering by column
  - First four bytes of 128-bit plaintext input occupy first column of in matrix
  - First four bytes of expanded key occupy first column of w matrix

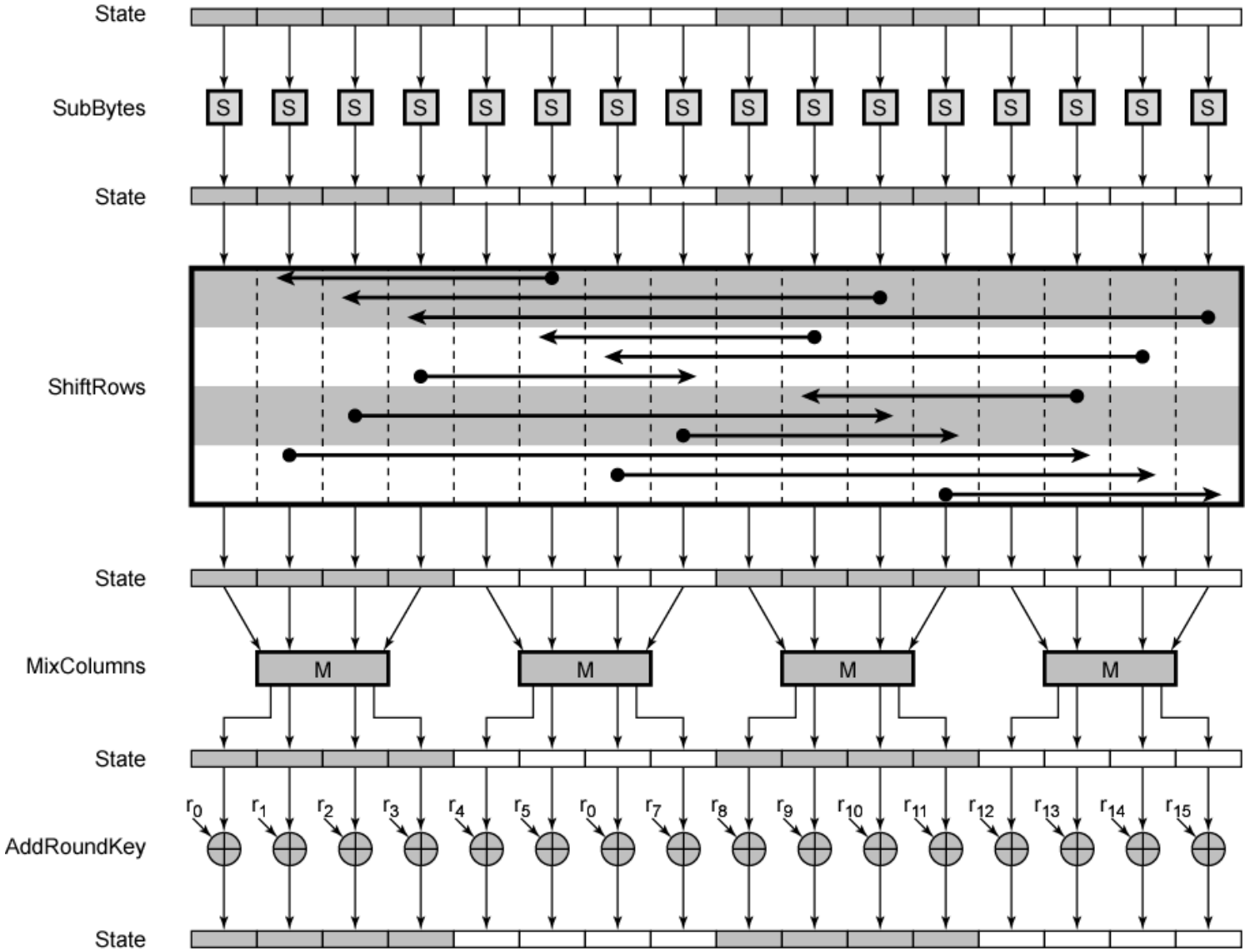
AES Encryption and Decryption



AES Comments (1)

- Key expanded into array of forty-four 32-bit words, w[i]
  - Four distinct words (128 bits) serve as round key for each round
- Four different stages
  - One permutation and three substitution
    - Substitute bytes uses S-box table to perform byte-by-byte substitution of block
    - Shift rows is permutation that performed row by row
    - Mix columns is substitution that alters each byte in column as function of all of bytes in column
    - Add round key is bitwise XOR of current block with portion of expanded key
- Simple structure
  - For both encryption and decryption, cipher begins with Add Round Key stage
  - Followed by nine rounds,
    - Each includes all four stages
  - Followed by tenth round of three stages

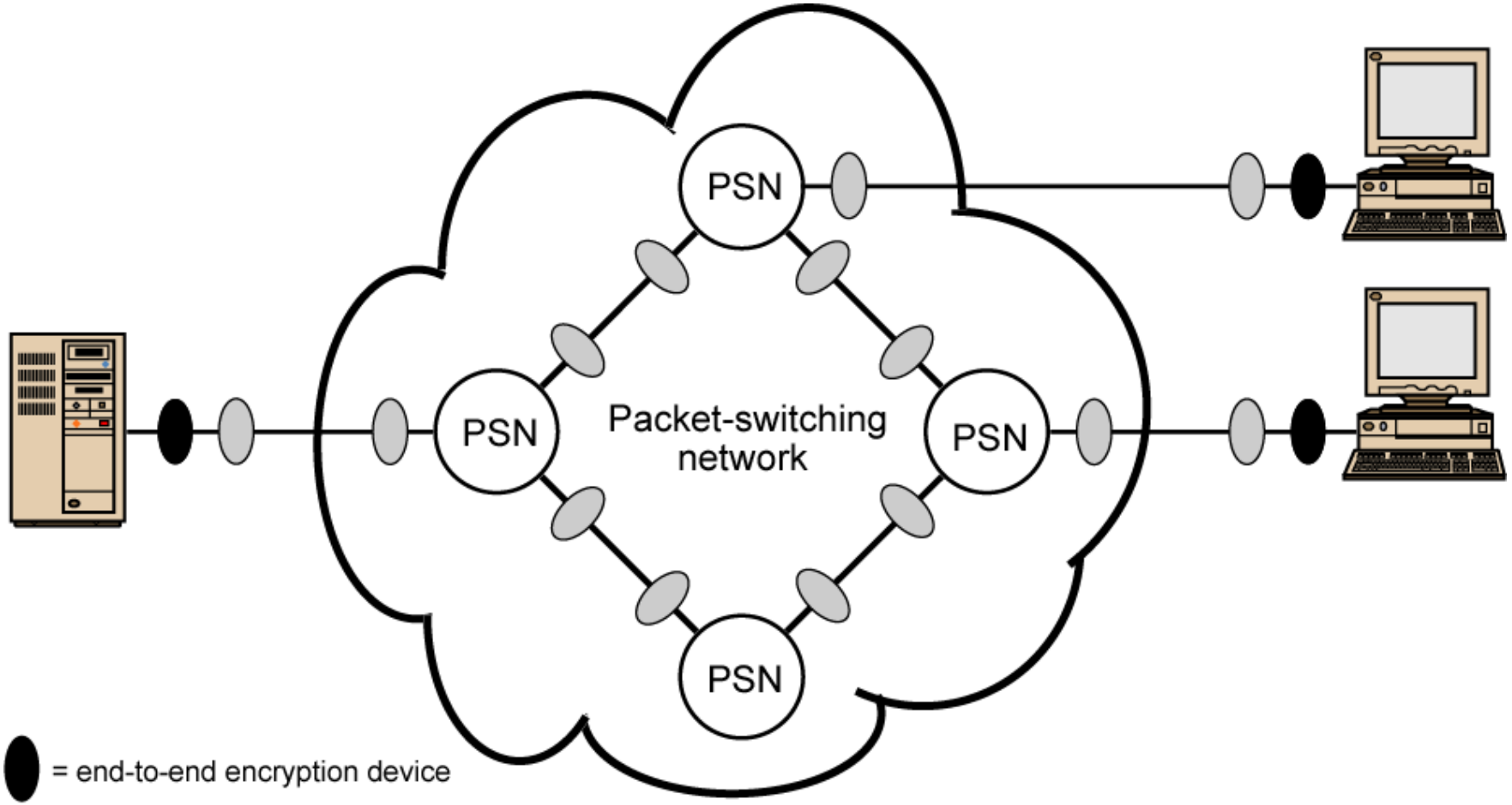
AES Encryption Round



AES Comments (2)

- Only Add Round Key stage uses key
  - Begin and ends with Add Round Key stage
  - Any other stage at beginning or end, reversible without key
    - Adds no security
- Add Round Key stage by itself not formidable
  - Other three stages scramble bits
  - By themselves provide no security because no key
- Each stage easily reversible
- Decryption uses expanded key in reverse order
  - Not identical to encryption algorithm
- Easy to verify that decryption does recover plaintext
- Final round of encryption and decryption consists of only three stages
  - To make the cipher reversible

Location of Encryption Devices



Link Encryption

- Each communication link equipped at both ends
- All traffic secure
- High level of security
- Requires lots of encryption devices
- Message must be decrypted at each switch to read address (virtual circuit number)
- Security vulnerable at switches
  - —Particularly on public switched network

End to End Encryption

- Encryption done at ends of system
- Data in encrypted form crosses network unaltered
- Destination shares key with source to decrypt
- Host can only encrypt user data
  - Otherwise switching nodes could not read header or route packet
- Traffic pattern not secure

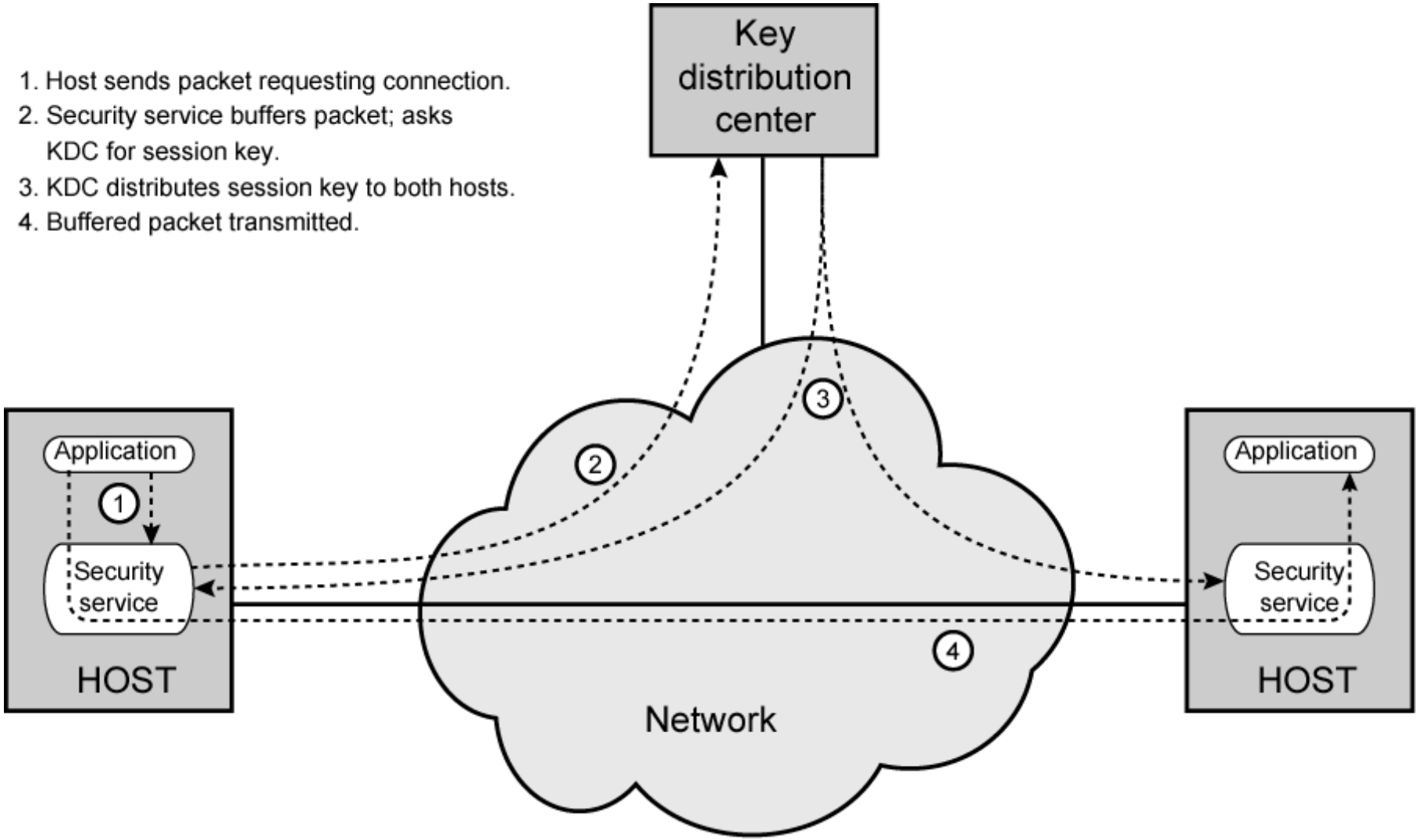
- Use both link and end to end

Key Distribution

- Key selected by A and delivered to B
- Third party selects key and delivers to A and B
- Use old key to encrypt and transmit new key from A to B
- Use old key to transmit new key from third party to A and B

Automatic Key Distribution (diag)

1. Host sends packet requesting connection.
2. Security service buffers packet; asks KDC for session key.
3. KDC distributes session key to both hosts.
4. Buffered packet transmitted.



Automatic Key Distribution

- Session Key
  - Used for duration of one logical connection
  - Destroyed at end of session
  - Used for user data
- Permanent key
  - Used for distribution of keys
- Key distribution center
  - Determines which systems may communicate
  - Provides one session key for that connection
- Security service module (SSM)
  - Performs end to end encryption
  - Obtains keys for host

Traffic Padding

- Produce cipher text continuously
- If no plain text to encode, send random data
- Make traffic analysis impossible

Message Authentication

- Protection against active attacks
  - Falsification of data
  - Eavesdropping
- Message is authentic if it is genuine and comes from the alleged source
- Authentication allows receiver to verify that message is authentic
  - Message has not altered
  - Message is from authentic source
  - Message timeline

Authentication Using Encryption

- Assumes sender and receiver are only entities that know key
- Message includes:
  - —error detection code
  - —sequence number
  - —time stamp

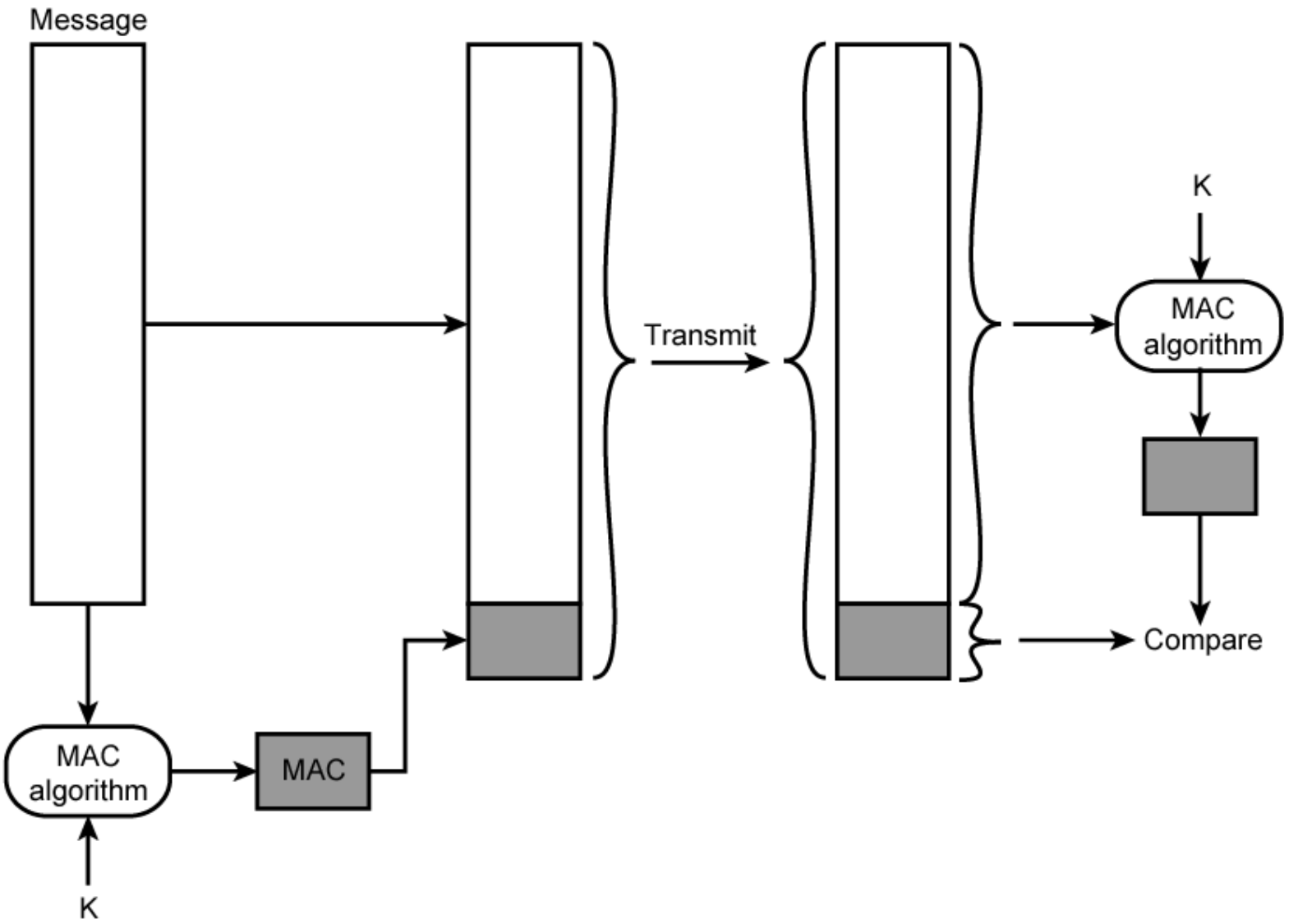
Authentication Without Encryption

- Authentication tag generated and appended to each message
- Message not encrypted
- Useful for:
  - —Messages broadcast to multiple destinations
    - Have one destination responsible for authentication
  - —One side heavily loaded
    - Encryption adds to workload
    - Can authenticate random messages
  - —Programs authenticated without encryption can be executed without decoding

Message Authentication Code

- Generate authentication code based on shared key and message
- Common key shared between A and B
- If only sender and receiver know key and code matches:
  - —Receiver assured message has not altered
  - —Receiver assured message is from alleged sender
  - —If message has sequence number, receiver assured of proper sequence

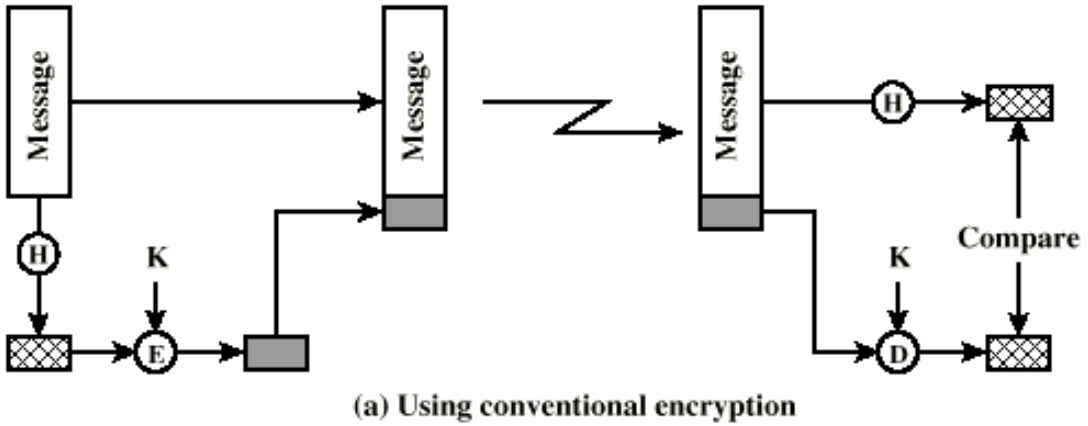
Message Authentication Using Message Authentication Code



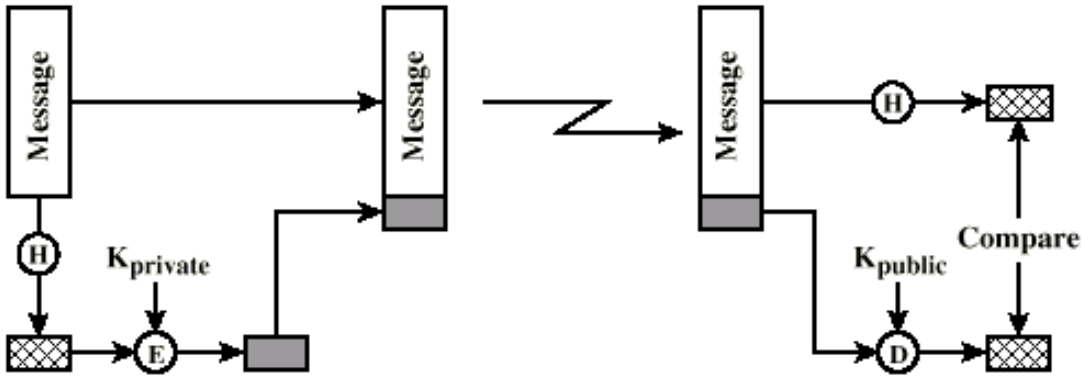
One Way Hash Function

- Accepts variable size message and produces fixed size tag (message digest)
- Advantages of authentication without encryption
  - Encryption is slow
  - Encryption hardware expensive
  - Encryption hardware optimized to large data
  - Algorithms covered by patents
  - Algorithms subject to export controls (from USA)

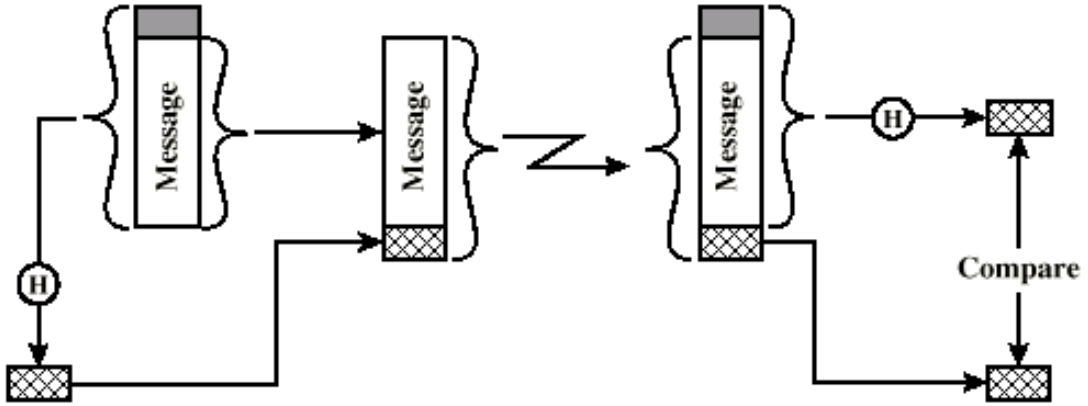
Using One Way Hash



(a) Using conventional encryption



(b) Using public-key encryption



(c) Using secret value

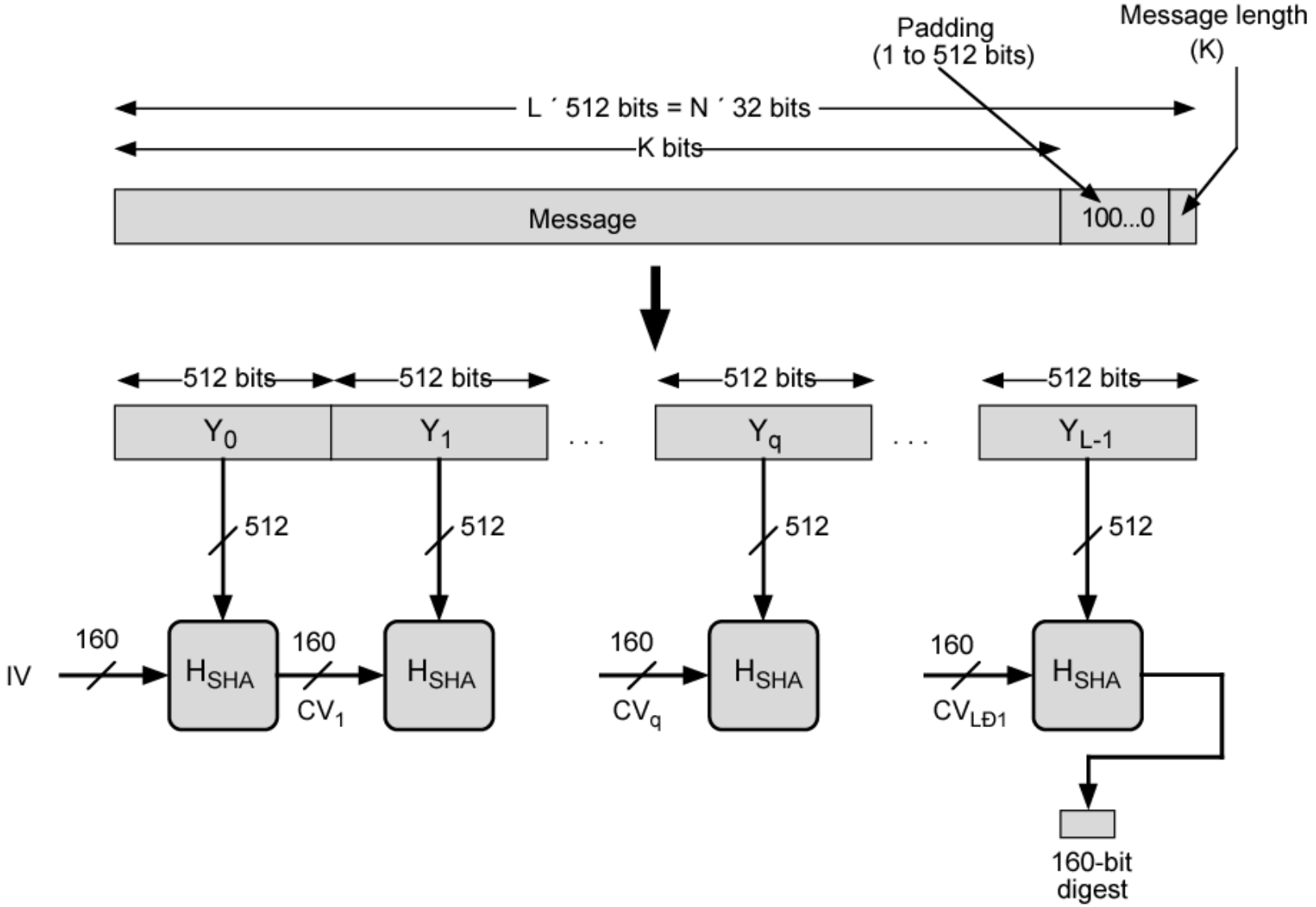
Secure Hash Functions

- Hash function must have following properties:
  - —Can be applied to any size data block
  - —Produce fixed length output
  - —Easy to compute
  - —Not feasible to reverse
  - —Not feasible to find two message that give the same hash

SHA-1

- Secure Hash Algorithm 1
- Input message less than 2^{64} bits
  - Processed in 512 bit blocks
- Output 160 bit digest

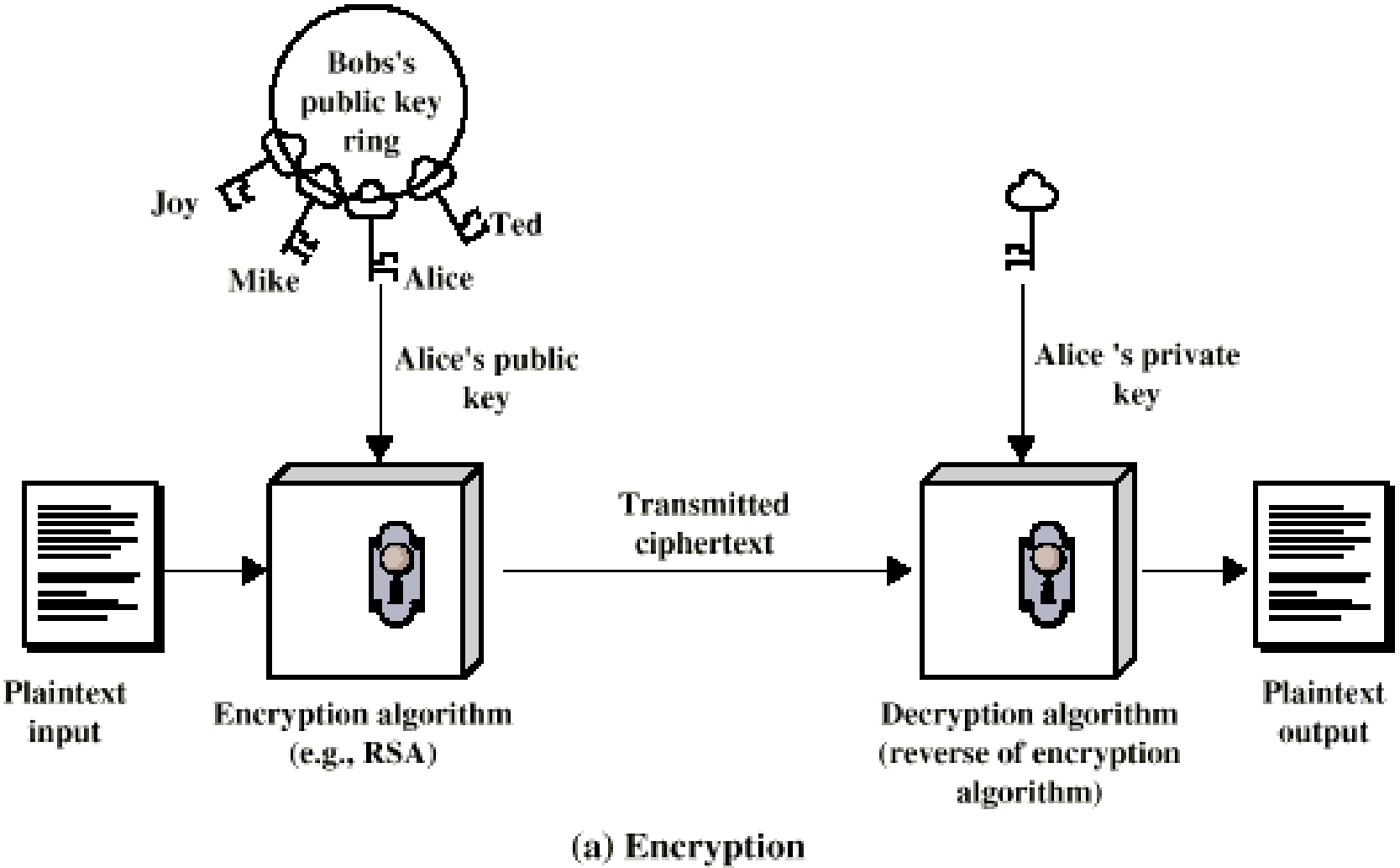
Message Digest Generation Using SHA-1



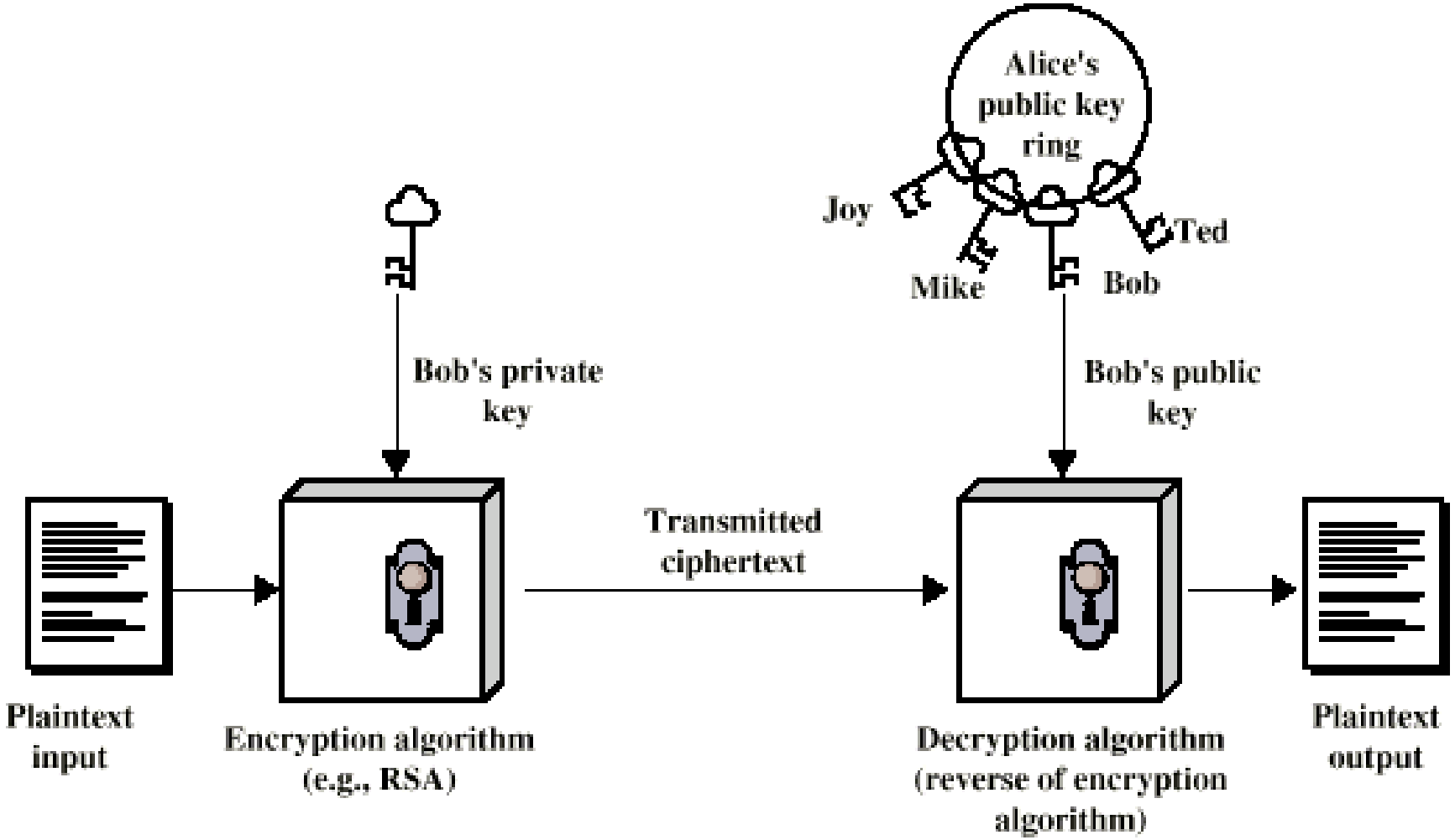
Public Key Encryption

- Based on mathematical algorithms
- Asymmetric
  - —Use two separate keys
- Ingredients
  - —Plain text
  - —Encryption algorithm
  - —Public and private key
  - —Cipher text
  - —Decryption algorithm

Public Key Encryption - Encryption



Public Key Encryption – Authentication



(b) Authentication

Public Key Encryption - Operation

- One key made public
  - —Used for encryption
- Other kept private
  - —Used for decryption
- Infeasible to determine decryption key given encryption key and algorithm
- Either key can be used for encryption, the other for decryption

Steps

- User generates pair of keys
- User places one key in public domain
- To send a message to user, encrypt using public key
- User decrypts using private key

Digital Signature

- Sender encrypts message with their private key
- Receiver can decrypt using sneders public key
- This authenticates sender, who is only person who has the matching key
- Does not give privacy of data
  - Decrypt key is public

RSA Algorithm

Key Generation

Select p, q	p and q both prime
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$
Calculate d	$d = e^{-1} \bmod \phi(n)$
Public key	$KU = \{e, n\}$
Private key	$KR = \{d, n\}$

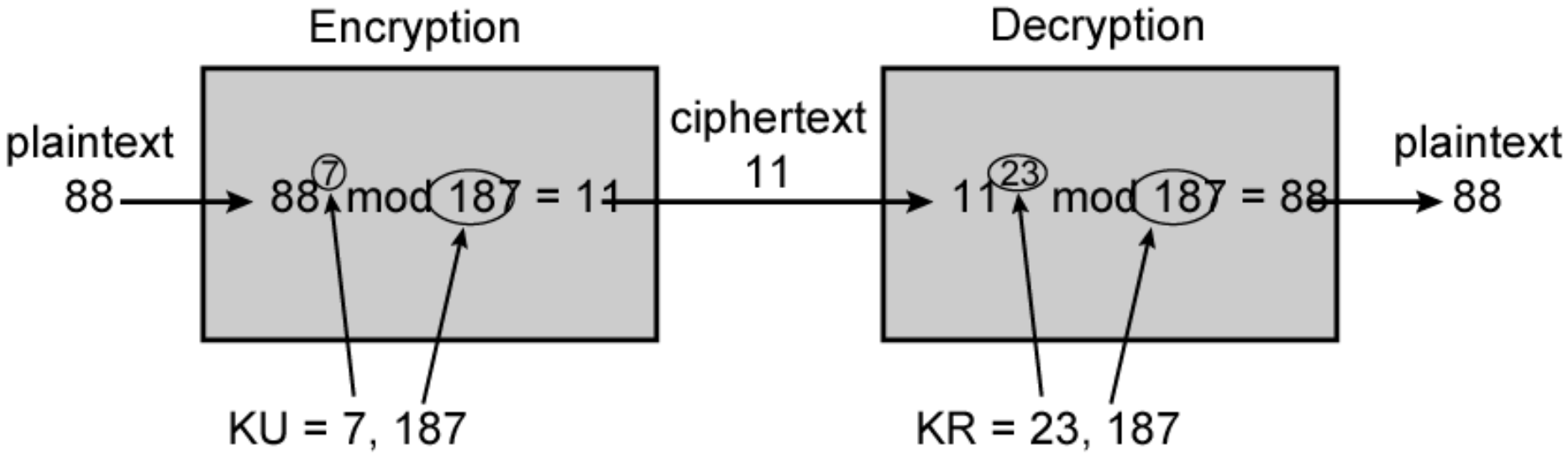
Encryption

Plaintext:	$M < n$
Ciphertext:	$C = M^e \pmod{n}$

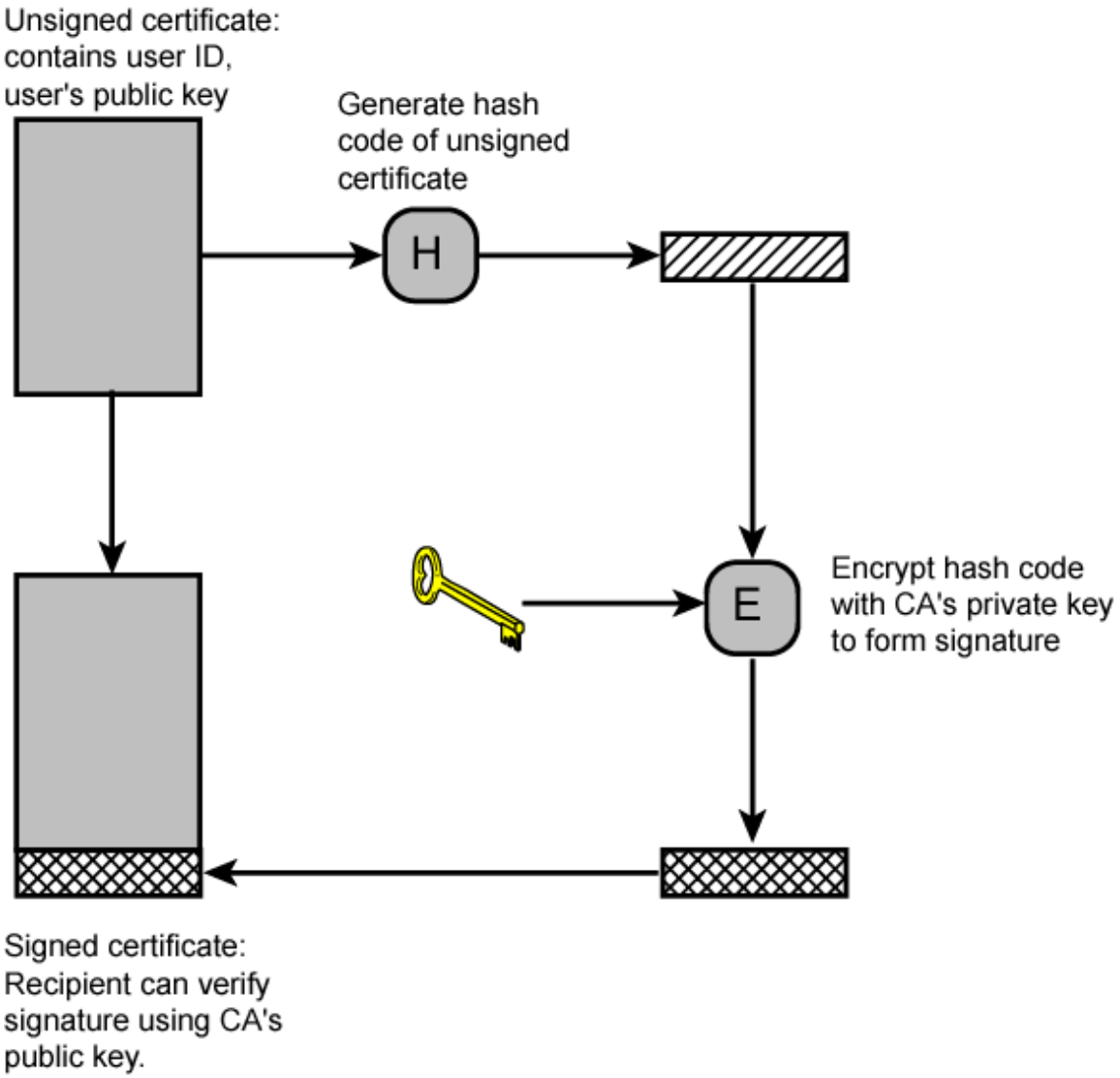
Decryption

Ciphertext:	C
Plaintext:	$M = C^d \pmod{n}$

RSA Example


Encryption
Decryption
plaintext
88
88^7 mod 187 = 11
ciphertext
11
11^23 mod 187 = 88
plaintext
88
KU = 7, 187
KR = 23, 187

Public Key Certificate Use



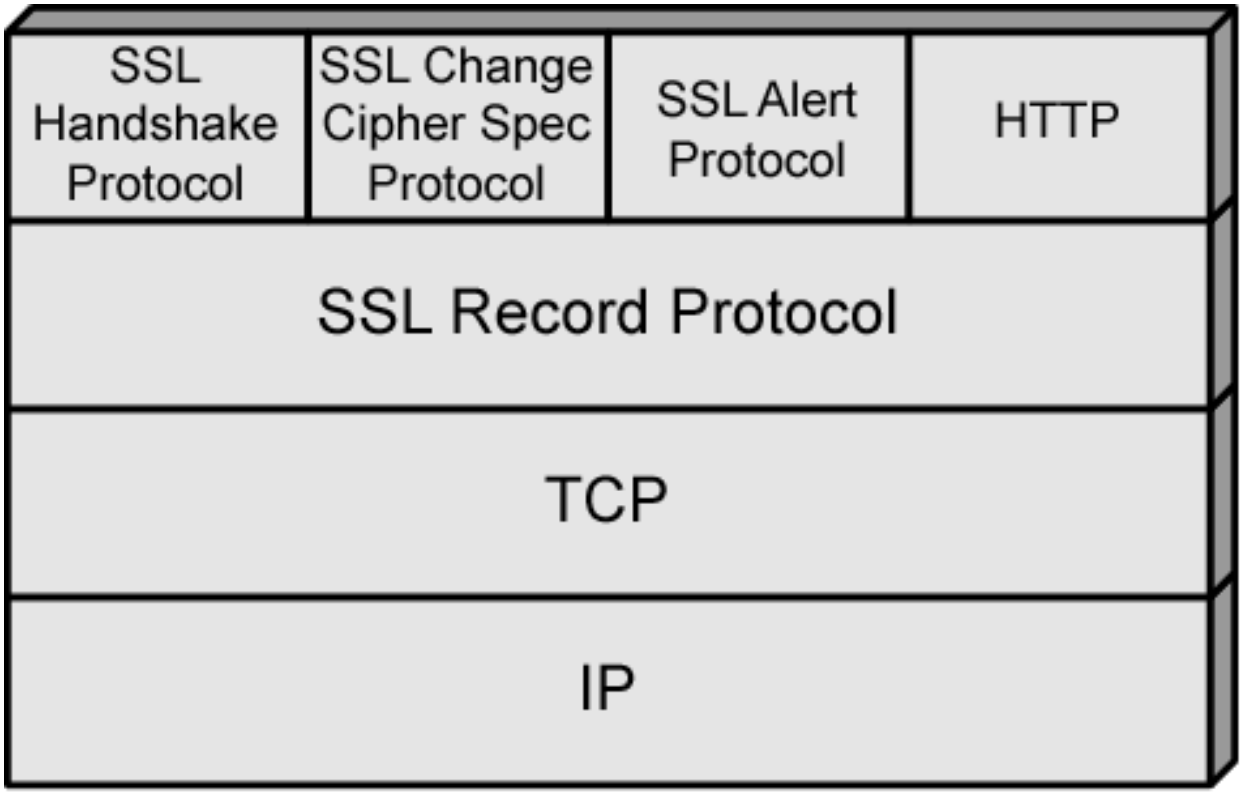
Secure Sockets Layer
Transport Layer Security

- Security services
- Transport Layer Security defined in RFC 2246
- SSL general-purpose service
  - —Set of protocols that rely on TCP
- Two implementation options
  - —Part of underlying protocol suite
    - Transparent to applications
  - —Embedded in specific packages
    - E.g. Netscape and Microsoft Explorer and most Web servers
- Minor differences between SSLv3 and TLS

SSL Architecture

- SSL uses TCP to provide reliable end-to-end secure service
- SSL two layers of protocols
- Record Protocol provides basic security services to various higher-layer protocols
  - In particular, HTTP can operate on top of SSL
- Three higher-layer protocols
  - Handshake Protocol
  - Change Cipher Spec Protocol
  - Alert Protocol
  - Used in management of SSL exchanges (see later)

SSL Protocol Stack



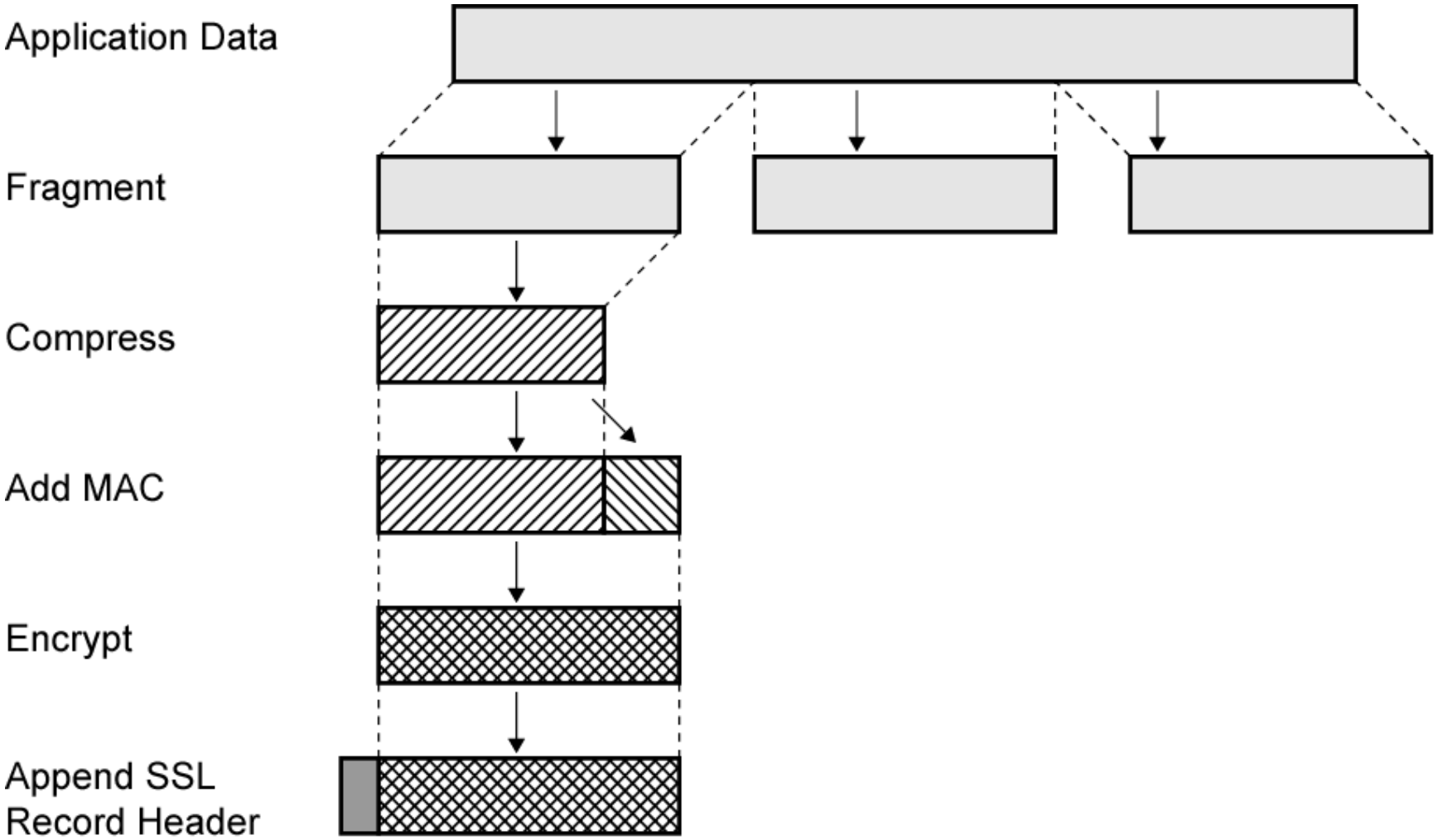
SSL Connection and Session

- Connection
  - Transport that provides suitable type of service
  - Peer-to-peer
  - Transient
  - Every connection associated with one session
- Session
  - Association between client and server
  - Created by Handshake Protocol
  - Define set of cryptographic security parameters
  - Used to avoid negotiation of new security parameters for each connection
- Maybe multiple secure connections between parties
- May be multiple simultaneous sessions between parties
  - Not used in practice

SSL Record Protocol

- Confidentiality
  - Handshake Protocol defines shared secret key
  - Used for symmetric encryption
- Message Integrity
  - Handshake Protocol defines shared secret key
  - Used to form message authentication code (MAC)
- Each upper-layer message fragmented
  - 2^{14} bytes (16384 bytes) or less
- Compression optionally applied
- Compute message authentication code
- Compressed message plus MAC encrypted using symmetric encryption
- Prepend header

SSL Record Protocol Operation



Record Protocol Header

- Content Type (8 bits)
  - change_cipher_spec, alert, handshake, and application_data
  - No distinction between applications (e.g., HTTP)
    - Content of application data opaque to SSL
- Major Version (8 bits) – SSL v3 is 3
- Minor Version (8 bits) - SSLv3 value is 0
- Compressed Length (16 bits)
  - Maximum 2^{14} + 2048
- Record Protocol then transmits unit in TCP segment
- Received data are decrypted, verified, decompressed, and reassembled and then delivered

Change Cipher Spec Protocol

- Uses Record Protocol
- Single message
  - —Single byte value 1
- Cause pending state to be copied into current state
  - —Updates cipher suite to be used on this connection

Alert Protocol

- Convey SSL-related alerts to peer entity
- Alert messages compressed and encrypted
- Two bytes
  - —First byte warning(1) or fatal(2)
    - If fatal, SSL immediately terminates connection
    - Other connections on session may continue
    - No new connections on session
  - —Second byte indicates specific alert
  - —E.g. fatal alert is an incorrect MAC
  - —E.g. nonfatal alert is close_notify message

Handshake Protocol

- Authenticate
- Negotiate encryption and MAC algorithm and cryptographic keys
- Used before any application data sent

Handshake Protocol – Phase 1 Initiate Connection

- Version
  - Highest SSL version understood by client
- Random
  - Client-generated random structure
  - 32-bit timestamp and 28 bytes from secure random number generator
  - Used during key exchange to prevent replay attacks
- Session ID
  - Variable-length
  - Nonzero indicates client wishes to update existing connection or create new connection on session
  - Zero indicates client wishes to establish new connection on new session
- CipherSuite
  - List of cryptographic algorithms supported by client
  - Each element defines key exchange algorithm and CipherSpec
- Compression Method
  - Compression methods client supports

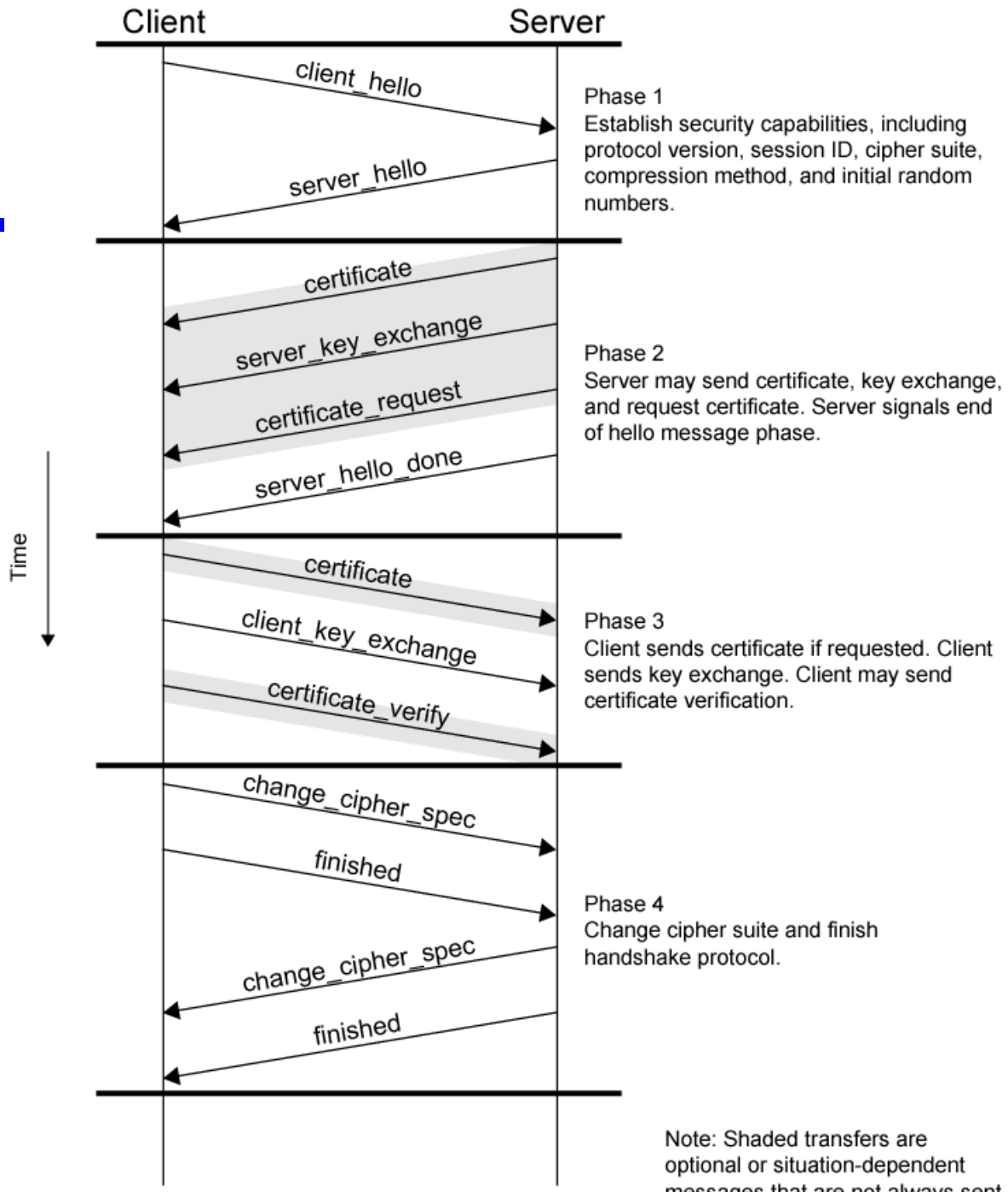
Handshake Protocol – Phase 2, 3

- Client waits for server_hello message
  - Same parameters as client_hello
- Phase 2 depends on underlying encryption scheme
- Final message in Phase 2 is server_done
  - Required
- Phase 3
  - Upon receipt of server_done, client verifies certificate if required and check server_hello parameters
  - Client sends messages to server, depending on underlying public-key scheme

Handshake Protocol – Phase 4

- Completes setting up
- Client sends change_cipher_spec
- Copies pending CipherSpec into current CipherSpec
  - Not considered part of Handshake Protocol
  - Sent using Change Cipher Spec Protocol
- Client sends finished message under new algorithms, keys, and secrets
- Finished message verifies key exchange and authentication successful
- Server sends own change_cipher_spec message
- Transfers pending to current CipherSpec
- Sends its finished message
- Handshake complete

Handshake Protocol Action



IPv4 and IPv6 Security

- IPSec
- Secure branch office connectivity over Internet
- Secure remote access over Internet
- Extranet and intranet connectivity
- Enhanced electronic commerce security

IPSec Scope

- Authentication header
- Encapsulated security payload
- Key exchange
- RFC 2401,2402,2406,2408

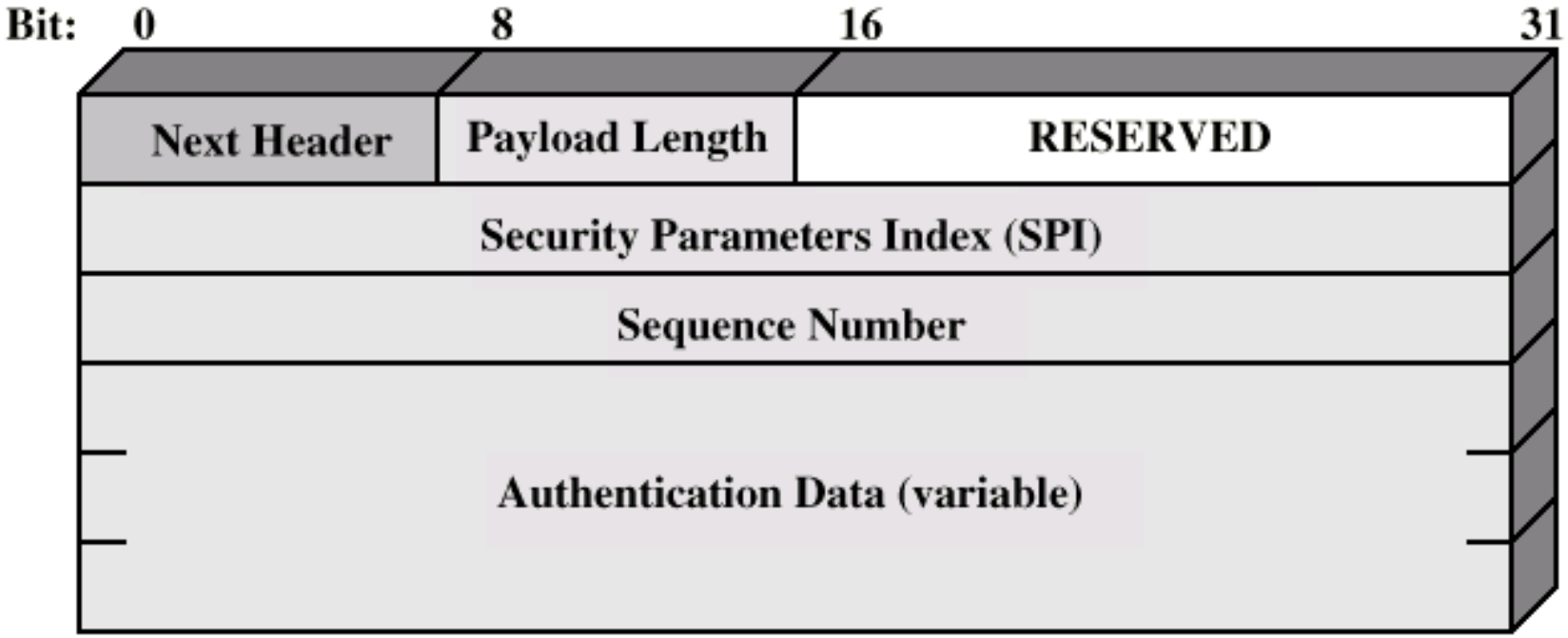
Security Association

- One way relationship between sender and receiver
- For two way, two associations are required
- Three SA identification parameters
  - —Security parameter index
  - —IP destination address
  - —Security protocol identifier

SA Parameters

- Sequence number counter
- Sequence counter overflow
- Anti-reply windows
- AH information
- ESP information
- Lifetime of this association
- IPSec protocol mode
  - —Tunnel, transport or wildcard
- Path MTU

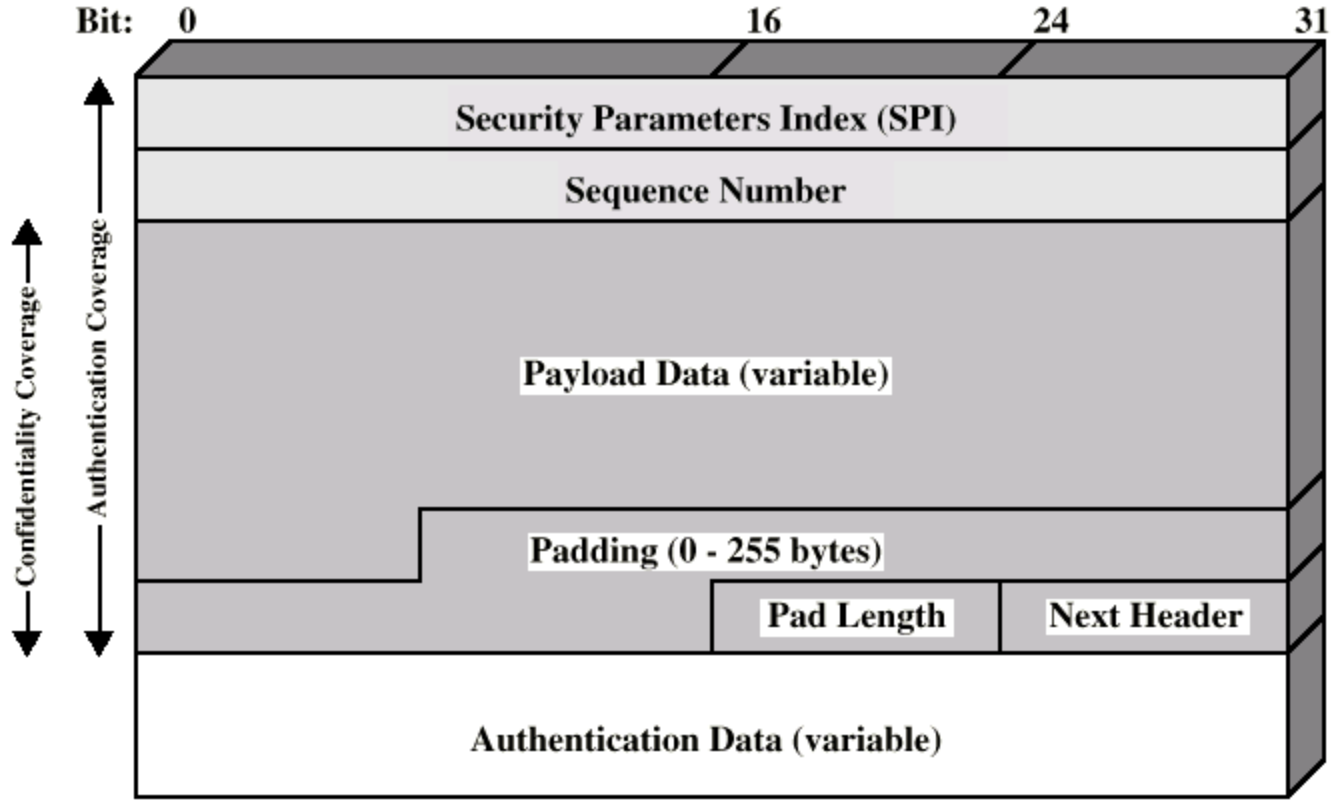
Authentication Header



Encapsulating Security Payload

- ESP
- Confidentiality services

ESP Packet


Bit:
0
16
24
31
Security Parameters Index (SPI)
Sequence Number
Payload Data (variable)
Padding (0 - 255 bytes)
Pad Length
Next Header
Authentication Data (variable)
Confidentiality Coverage
Authentication Coverage

Required Reading

- Stallings chapter 21
- Web sites on public/private key encryption
- RFCs mentioned
  - www.rfc-editor.org